Information Sciences 334–335 (2016) 323–337

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An efficient algorithm for attribute-based subsequence matching

Jun-Feng Qu*, Lei Yuan, Yannong Huang, Zhao Wu

School of Mathematics and Computer Science, Hubei University of Arts and Science, Xiangyang 441053, China

ARTICLE INFO

Article history:
Received 27 November 2014
Revised 4 October 2015
Accepted 8 December 2015
Available online 12 December 2015

MSC:
00-01
99-00

Keywords:
Subsequence matching
Attribute-based sequence

ABSTRACT

Subsequence matching plays a fundamental role in the solutions to sequence-related problems such as sequence classification and similarity search. Subsequence matching is to find out expected event sequences from a database with a specified sequence composed of events. One cannot specify some or all of the events in the sequence but can give attribute values of these events in some applications. Therefore, a problem is how to find out expected event sequences from a database with an attribute-based sequence composed of attribute values. We propose an algorithm for the problem. The algorithm uses a list structure to store the temporal information about both sequences and events associated with values in a database. The structure can be compressed using a simple technique. Experimental data show that the proposed algorithm is one to two orders of magnitude faster than the state-of-the-art algorithm, especially for real-world databases, and that the technique is effective.

1. Introduction

Various sequence data, such as transaction record, DNA sequence, network log, are generated in quantity every day. Researches based on these data include sequence classification [16], sequence clustering [4], frequent sequence mining [2,17,18], similarity search [8,10], sequence alignment [9,13,14], motif discovery [3,6], and so on. These researches involve a common fundamental operation, namely subsequence matching.

A sequence is an ordered list of events. Given two sequences s and s', s is a subsequence of s' if (1) each event in s is in s' and (2) the order in which all the events in s appear is the same as the order in which these events appear in s'. Given a sequence database DB and a sequence s, the subsequence matching problem is to discover from DB all the sequences each of which s is a subsequence. For example, on a movie-on-demand website, each user is associated with a sequence in which all the events are titles sorted in order of the click date of related movies. A database stores the sequences of all users. One can perform a subsequence matching algorithm on the database with sequence $<$ "Titanic", "The Aviator" $>$ to find out the users who watched movie "Titanic" and subsequently "The Aviator".

Consider several real-world applications. The first is to recommend a fantasy movie, in which the potential audience are defined as the users who watched at least three fantasy movies. The second application is to create a discussion group about "Ang Lee", in which the potential members are defined as the users who watched at least two movies directed by "Ang Lee". The third is to study how movie "Titanic" creates star "Leonardo DiCaprio" [12], in which one needs to find out the users who watched "Titanic" and subsequently one or more movies acted by "Leonardo DiCaprio".

* Corresponding author. Tel.: +86 07103593152.
E-mail addresses: qmxwt@163.com (J.-F. Qu), 1390405958@qq.com (L. Yuan), yannonghuang@gmail.com (Y. Huang), wuzhao73@163.com (Z. Wu).

http://dx.doi.org/10.1016/j.ins.2015.12.004
0020-0255/

Table 1
The sequence table of a sample database.

Sequence					
<	B,	A,	C	>	
<	E,	A,	C,	D	>
<	D,	C,	B,	C	>
<	A,	B	>		
<	A,	E,	B	>	
<	E,	D,	C,	A	>

Table 2
The attribute table of a sample database.

Event	Attribute value			
A	t,	u,	x	
B	t,	u,	z	
C	t,	x,	y	
D	t,	y		
E	t,	u,	x,	z

A method based on a subsequence matching algorithm can be used in these applications. For example, the potential audience of a fantasy movie can be found out as follows. Firstly, a title set of all the movies labeled "fantasy" is created. Secondly, the subsequence matching algorithm is performed on the database with sequences, each of which is composed of three titles from the set. Suppose the cardinality of the set is n, and then the algorithm has to be performed n^3 times. Therefore, the method is not efficient.

Generally, an event is associated with a number of attribute values in real-world databases. We are required to specify a sequence when performing a subsequence matching task. However, it is expected that some or all of the events in the sequence can be replaced with their attribute values in some cases. For example, a movie title is associated with the genres, directors, actresses, and actors of the related movie, and it is more straightforward to find out the users with sequences < "fantasy", "fantasy", "fantasy" >, < "Ang Lee", "Ang Lee" >, and < "Titanic", "Leonardo DiCaprio" > for the above applications. Such sequences involving attribute values are called attribute-based sequences.

The attribute-based subsequence matching problem is to find out excepted sequences from a database with a given attribute-based sequence. The problem was first proposed in [12]. Note that an event sequence can be regarded as an attribute-based sequence because an event can be regarded as a particular attribute value of the event itself. Therefore, the problem is a generalization of the subsequence matching problem.

In this study, we solve the attribute-based subsequence matching problem. Firstly, a novel seid-list (Sequence & Event Identifier List) structure is proposed. Seid-lists generated from a database can store the temporal information about both sequences and events associated with values. We further introduce a simple technique to generate a compressed seid-list structure. Secondly, a fast algorithm called LIA (seid-List Intersection Algorithm) is developed. LIA performs an attribute-based subsequence matching task by seid-list intersections. Thirdly, extensive experiments on various databases were performed to compare LIA with the state-of-the-art algorithms. Experimental results are reported and discussed.

The rest of this paper is organized in the following way. After the background is stated in Section 2, we introduce the seid-list structure and the LIA algorithm in Section 3. Section 4 presents the correctness proof and complexity analysis of LIA. Section 5 reports experimental results which are discussed in Section 6. Finally, concluding remarks are given in Section 7.

2. Background

This section gives the formal description of the attribute-based subsequence matching problem and subsequently introduces related work.

2.1. Problem definition

Let E be a set of events and V be a set of attribute values. A database DB is composed of a sequence table and an attribute table. A sequence table is composed of sequences, each of which is an ordered list of events from E. In an attribute table, each event from E is associated with attribute values from V. Table 1 shows a sequence table, in which E is {A, B, C, D, E}. Table 2 shows an attribute table, in which V is {t, u, x, y, z}. The two tables constitute a sample database. An attribute-based sequence is an ordered list of attribute values from V.

Definition 1. Given an attribute-based sequence q in form of $< v_1, v_2, \ldots, v_x >$ and an event sequence s in form of $< e_1, e_2, \ldots, e_y >$, s is matched by q (or q matches s) iff there are events $e_{i_1}, e_{i_2}, \ldots, e_{i_x}$ in s such that e_{i_k} is associated with v_k $(1 \le k \le x)$ and $1 \le i_1 < i_2 < \cdots < i_x \le y$.

Definition 2. Given a database DB and an attribute-based sequence q, the attribute-based subsequence matching problem is to discover from DB all the sequences which are matched by q.

Consider an attribute-based sequence < z, x, y > . It matches the first sequence in Table 1, because events B, A, and C are associated with values z, x, and y, respectively. It also matches the second sequence, but it does not match the others.

2.2. Related work

Subsequence matching is generally a fundamental operation in many applications and can be defined in diverse forms.